WE CLAIM:

1. An encoder adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements, the encoder comprising a data organization component, a linear state sequencer and a state-to-data-elements converter, wherein:

the data organization component is adapted to receive the sequence of source data elements and to receive a sequence of state-derived data elements from the state-to-data-elements converter and to output a data organization output sequence which includes every data element of the sequence of source data elements and which on an ongoing basis includes inserted data elements, each inserted data element inserted at a given time instant being either: i) one state-derived data element being output by the state-to-data elements converter at the given time instant or ii) a sum of one state-derived data element being output by the state-to-data elements converter at the given time instant and a linear combination of source data elements being output by the data organization component at the given time instant;

the linear state sequencer being adapted to maintain a state consisting of state data elements and to perform linear state sequencing as a function of the data organization output sequence, which is provided as input to the linear state sequencer, and as a function of the state data elements;

the state-to-data-elements converter being adapted to produce said sequence of state-derived data elements, wherein each state-derived data element is a linear combination of the state data elements;

wherein the linear state sequencer is configured to satisfy the following specifications:

a) if the linear state sequencer state is zero at a time i, any non-zero data organization output sequence data element at time i will result in a non-zero state at time $i+1$;

b) if the linear state sequencer state is zero at a time i, non-zero at time $i+1$, but again zero at a later time $k > i+1$, then necessarily there must be a non-zero data organization output sequence data element at some time j, with $i < j < k$;

and wherein the data organization component and the state-to-data-elements converter are configured in relation to the linear state sequencer to satisfy the following specifications:

c) if the linear state sequencer state is non-zero at a time x, non-zero at a later time $z > x$, and non-zero for all times between x and z, the time z cannot be advanced indefinitely, in so doing increasing the duration of the time interval $[x,z]$ during which the linear state sequencer state is always non-zero, without necessitating a non-zero data organization output sequence data element at some time y, with $x \leq y < z$;

d) data element insertion by the data organization component into the sequence of source data elements to produce the data organization output sequence does not render any linear state sequencer state unreachable;

the sequence of primary coded data elements being equal to the data organization output sequence.

2. An encoder according to claim 1 wherein the data organization component is adapted to insert inserted data elements on a periodic or pseudo-periodic basis.

3. A composite code encoder comprising:

an encoder according to claim 1;

a re-ordering function adapted to produce a re-ordered version of the sequence of primary coded data elements;

a RSC (recursive systematic convolutional) encoder adapted to receive as input the re-ordered version of the sequence of primary coded data elements and to produce a sequence of coded data elements.

4. A composite code encoder adapted to encode a sequence of source data elements to produce a first sequence of primary coded data elements which satisfy a first set of constraints equivalent to the encoder of claim 1, and which after being re-ordered to form a second sequence of coded data elements, satisfy a second set of constraints of another code.

5. A composite code encoder adapted to encode a sequence of source data elements to produce a first sequence of primary coded data elements which satisfy a first set of constraints equivalent to the encoder of claim 1, and which after being re-ordered to form a second sequence of coded data elements, satisfy a second set of constraints that is equivalent to another encoder of claim 1.

6. A composite code encoder according to claim 4 wherein the second set of constraints is equivalent to the first set of constraints.

7. A composite code encoder according to claim 4 wherein the second set of constraints are those of a recursive systematic convolutional code.

8. A composite code encoder adapted to encode a sequence of source data elements to produce a first sequence of primary coded data elements which satisfy a first set of constraints equivalent to the encoder of claim 1, and which are such that after being re-ordered to form a plurality of other sequences of coded data elements, with each primary coded data element of the first sequence occurring in at least one of the plurality

of other sequences of coded data elements, each other sequence of coded data elements satisfies a respective set of constraints of a respective code.

9. An encoder adapted to encode a sequence of source data elements to produce a sequence of coded data elements, wherein a self-interlocking sequence that is an ordering of the coded data elements that includes each coded data element at least twice satisfies a set of constraints equivalent to those satisfied by the sequence of primary coded data elements of claim 1.

10. An encoder comprising encoding circuitry adapted to implement a set of constraints equivalent to the encoder of claim 1.

11. An encoder according to claim 1 wherein the state sequencer is an N state sequencer with N = 4, 8, 16 or 32.

12. An encoder according to claim 1 further adapted to produce auxiliary coded data elements which are linear combinations of the state data elements and the primary coded data elements.

13. An encoder according to claim 4 further adapted to produce auxiliary coded data elements which are linear combinations of the state data elements and the primary coded data elements.

14. A method of generating an interleaver for use in the encoder of claim 5 defining how the first sequence of primary coded data elements is re-ordered to form the second sequence of primary coded data elements, the method comprising repeating the following steps until the entire interleaver is defined:

a) randomly generating a pair of indices which are not already included in the interleaver which will indicate where a random element of the first sequence of primary coded

data elements will end up in the second sequence of primary coded data elements;

b) performing one or more performance tests, the performance tests taking into account that some linear state sequencer state transitions are not possible for state transition intervals involving inserted data elements;

c) if the performance tests pass, adding the pair of indices to the interleaver; and

d) removing some indices previously added to the interleaver if no possible pair of indices which have not already been included in the interleaver passes the tests.

15. An encoder adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements, the encoder comprising a data insertion component and a linear state sequencer having state data elements, wherein:

the data insertion component is adapted to receive the sequence of source data elements and to output the sequence of primary coded data elements which includes every data element of the sequence of source data elements and which on an ongoing basis includes inserted data elements, each inserted data element having a linear dependence on the state data elements;

the linear state sequencer is adapted to perform linear state sequencing as a function of the sequence of primary coded data elements which is provided as input to the linear state sequencer and as a function of the state data elements; and

inserted data element insertion by the data insertion component into the sequence of source data elements to produce the sequence of primary coded elements does not render unreachable any particular set of values for the state data elements of the linear state sequencer.

16. An encoder adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements which satisfy a set of constraints equivalent to the encoder of claim 1.

17. An encoder adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements which satisfy a set of constraints equivalent to the encoder of claim 11.

18. An encoder adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements and a sequence of auxiliary coded data elements, the sequence of primary coded data elements together with the sequence of auxiliary coded data elements satisfying a set of constraints equivalent to the encoder of claim 12.

19. An encoder adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements which satisfy a set of constraints equivalent to the encoder of claim 15.

20. An encoder comprising encoding circuitry adapted to implement a set of constraints equivalent to the encoder of claim 15.

21. A soft-in soft-out decoder adapted to perform soft-in soft-out decoding in a manner consistent with the encoder of claim 1 of a first sequence of multi-valued probabilistic quantities to produce a second sequence of multi-valued probabilistic quantities, wherein the decoder is adapted to consider all linear state sequencer state transitions for state transition intervals without inserted data elements, and is adapted to consider for state transition intervals with inserted data elements only state transitions which are possible given that the inserted data elements have a predetermined dependency on the state.

22. A soft-in soft-out decoder adapted to perform soft-in soft-out decoding in a manner consistent with the encoder of claim 15 of a first sequence of multi-valued probabilistic quantities to produce a second sequence of multi-valued probabilistic quantities, wherein the decoder is adapted to consider all linear state sequencer state transitions for state transition intervals without inserted data elements, and is adapted to consider for state transition intervals with inserted data elements only state transitions which are possible given that the inserted data elements have a predetermined dependency on the state.

23. A soft-in soft-out decoder adapted to perform soft-in soft-out decoding, in accordance with a set of constraints equivalent to the encoder of claim 1, of multi-valued probabilistic quantities representing source data elements coded to satisfy said set of constraints, to produce a second sequence of multi-valued probabilistic quantities, wherein the decoder is adapted to consider all linear state sequencer state transitions for state transition intervals without inserted data elements, and is adapted to consider for state transition intervals with inserted data elements only state transitions which are possible given that the inserted data elements have a predetermined dependency on the state.

24. A soft-in soft-out decoder adapted to perform soft-in soft-out decoding, in accordance with a set of constraints equivalent to the encoder of claim 15, of multi-valued probabilistic quantities representing source data elements coded to satisfy said set of constraints to produce a second sequence of multi-valued probabilistic quantities, wherein the decoder is adapted to consider all linear state sequencer state transitions for state transition intervals without inserted data elements, and is adapted to consider for state transition intervals with inserted data elements only state transitions

which are possible given that the inserted data elements have a predetermined dependency on the state.

25. An iterative decoder adapted to perform iterative decoding of a sequence of multi-valued probabilistic quantities to produce a sequence of decoded data elements, the iterative decoder comprising a soft-in soft-out decoder according to claim 21.

26. An iterative decoder adapted to perform iterative decoding of a sequence of multi-valued probabilistic quantities to produce a sequence of decoded data elements, the iterative decoder comprising a soft-in soft-out decoder according to claim 22.

27. A signal comprising a sequence of primary coded data elements of a first code embodied on a transmission medium or a storage medium containing every data element from a sequence of source data elements, the sequence of primary coded data elements also containing on an ongoing basis inserted data elements, the inserted data elements having a linear dependence upon a state, the state being determined by performing linear state sequencing as a function of the sequence of primary data elements which is provided as input to the linear state sequencing wherein the sequence of primary coded data elements satisfies the following specifications:

a) if the linear state sequencing state is zero at a time i, any non-zero data element of the sequence of primary coded data elements at time i will result in a non-zero state at time i+1;

b) if the linear state sequencing state is zero at a time i, non-zero at time i+1, but again zero at a later time k > i+1, then necessarily there must be a non-zero data element of the sequence of primary coded data elements at some time j, with i < j < k;

c) if the state of the linear state sequencing is non-zero at a time x, non-zero at a later time $z > x$, and non-zero for all times between x and z, the time z cannot be advanced indefinitely, in so doing increasing the duration of the time interval [x,z] during which the linear state sequencing state is always non-zero, without necessitating a non-zero data element of the sequence of primary coded data elements at some time y, with $x \leq y < z$; and

d) data element insertion into the sequence of source data elements to produce the sequence of primary coded data elements does not render any linear state sequencing states unreachable.

28. A signal according to claim 27 wherein the sequence of primary coded data elements is further adapted to, after being reordered, satisfy a set of constraints imposed by a second code.

29. A signal according to claim 28 wherein the set of constraints imposed by the second code is equivalent to a set of constraints imposed by another first code according to claim 28.

30. A signal according to claim 28 wherein the set of constraints imposed by the second code is equivalent to the set of constraints imposed by the first code.

31. A soft-in soft-out decoder adapted to perform soft-in soft-out decoding, in accordance with a set of constraints satisfied by the sequence of primary coded data elements of claim 27, of multi-valued probabilistic quantities representing said source data elements coded to satisfy said set of constraints, to produce a second sequence of multi-valued probabilistic quantities, wherein the decoder is adapted to consider all linear state sequencer state transitions for state transition intervals without inserted data elements, and is

adapted to consider for state transition intervals with inserted data elements only state transitions which are possible given that the inserted data elements have a predetermined dependency on the state.

32. A decoder adapted to iteratively decode quantities representative of a signal according to claim 28.

33. A decoder adapted to iteratively decode quantities representative of a signal according to claim 29.

34. A decoder according to claim 33 adapted to repeatedly, alternating between SISO decoding the first and second code, SISO decode one of the codes taking into account intrinsic information associated with the sequence of primary coded data elements associated with the one of the codes and taking into account extrinsic information generated by previously SISO decoding the other of the codes, to produce extrinsic information to be made available when next SISO decoding the other of the codes.

35. A method of stopping an iterative decoder decoding a composite code comprising at least two constituent codes, a partial iteration of the iterative decoder comprising performing SISO decoding of one of the constituent codes, the method comprising:

checking three conditions as follows for each multiple instance data element:

a) after each partial iteration a change in an extrinsic associated with each instance of a data element, not including a next instance to undergo SISO processing, must not disagree with a decision associated with this same instance;

b) decisions must agree between all instances of a data element;

c) decisions must be unambiguous;

and when the three conditions are satisfied, stopping the iterative decoder from performing any further partial iterations.

36. A method according to claim 35 adapted for application wherein said at least two constituent codes comprise two constituent codes.

37. A processing platform readable medium having stored thereon code means executable by a processing platform, the code means when executed being adapted to encode a sequence of source data elements to produce a sequence of primary coded data elements according to a first code, the code means comprising functionality equivalent to a data organization component, a linear state sequencer and a state-to-data-elements converter, wherein:

the data organization is adapted to receive the sequence of source data elements and to receive a sequence of state-derived data elements from the state-to-data-elements converter to output a data organization output sequence which includes every data element of the sequence of source data elements and which on an ongoing basis includes inserted data elements, each inserted data element inserted at a given time instant being either: i) one state-derived data element being output by the state-to-data elements converter at the given time instant or ii) a sum of one state-derived data element being output by the state-to-data elements converter at the given time instant and a linear combination of source data elements being output by the data organization component at the given time instant;

the linear state sequencer being adapted to maintain a state consisting of state data elements and to perform linear state sequencing as a function of the data organization output sequence, which is provided as input to the linear state sequencer, and as a function of the state data elements;

the state-to-data-elements converter being adapted to produce said sequence of state-derived data elements, wherein each state-derived data element is a linear combination of the state data elements;

and wherein the data organization component and the state-to-data-elements converter are configured in relation to the linear state sequencer such that data element insertion by the data organization component into the sequence of source data elements to produce the data organization output sequence does not render any linear state sequencer state unreachable;

the sequence of primary coded data elements being equal to the data organization output sequence;

wherein the sequence of primary coded data elements after being reordered also satisfies a second set of constraints equivalent to a second code.

38. A processing platform readable medium according to claim 37 wherein the constraints of the second code are equivalent to those of the first code.

39. A processing platform readable medium according to claim 37 wherein the state sequencer is an N state sequencer with N = 4, 8, 16 or 32.

40. A processing platform readable medium having stored thereon code means executable by a processing platform, the code means when executed being adapted to repeatedly, alternating between SISO decoding the first and second code claim of 37, SISO decode one of the codes taking into account intrinsic information associated with the sequence of coded data elements associated with the one of the codes and taking into account extrinsic information generated by previously SISO decoding the other of the codes, to produce extrinsic information to be made available when next SISO decoding the other of the codes.

41. A processing platform readable medium having stored thereon code means executable by a processing platform, the code means when executed being adapted to repeatedly, alternating between SISO decoding the first and second code of claim of 38, SISO decode one of the codes taking into account intrinsic information associated with the sequence of coded data elements associated with the one of the codes and taking into account extrinsic information generated by previously SISO decoding the other of the codes, to produce extrinsic information to be made available when next SISO decoding the other of the codes.

42. A processing platform readable medium having stored thereon code means executable by a processing platform, the code means when executed being adapted to perform soft-in soft-out decoding in a manner consistent with the encoder of claim 1 of a first sequence of multi-valued probabilistic quantities to produce a second sequence of multi-valued probabilistic quantities, wherein the code means is adapted to consider all linear state sequencer state transitions for state transition intervals without inserted data elements, and is adapted to consider for state transition intervals with inserted data elements only state transitions which are possible given that the inserted data elements have a predetermined dependency on the state.